BAB VI. STATEMENT CONTROL

A. Statement IF

Seperti halnya Pascal, perintah IF dalam C++ juga digunakan untuk menyatakan pernyataan kondisional (bersyarat).

Sintaks sederhana IF adalah

```
if (kondisi)
      statement;
```

Statement pada sintaks di atas akan dilakukan jika kondisinya bernilai TRUE (tidak sama dengan nol).
Apabila statement yang akan dilakukan lebih dari satu, maka sintaksnya menjadi

```
if (kondisi)
      {
      statement1;
      statement2;
      .
      .
      }
```

Contoh sederhana penggunaan IF adalah untuk menentukan boleh tidaknya seseorang melihat film bioskop. Seseorang diperbolehkan menonton jika usianya 17 tahun ke atas. Berikut ini program C++ nya:

```
#include <iostream.h>
#include <conio.h>

void main()
{
      int usia;
      clrscr();
      cout << "Berapa usia Anda : ";
      cin >> usia;
      if (usia < 17)
            cout << "Anda tidak boleh menonton bioskop";
}
```

Statement IF juga dapat ditambahkan ELSE sebagai konsekuensi alternatif jika kondisi tidak dipenuhi (FALSE). Sintaksnya:

```
if (kondisi)
{
statement1;
statement2;
.
.
}
else  {
      statement1;
```

```
      statement2;
      }
```

Anda dapat modifikasi program C++ untuk menentukan boleh tidaknya seseorang menonton bioskop seperti di bawah ini:

```
#include <iostream.h>
#include <conio.h>

void main()
{
      int usia;
      clrscr();
      cout << "Berapa usia Anda : ";
      cin >> usia;
      if (usia < 17)
            cout << "Anda tidak boleh menonton bioskop";
      else cout << "Anda boleh menonton bioskop";
}
```

Untuk menyatakan kondisi (syarat) yang akan dicek pada IF, Anda dapat menggunakan operator logika dan operator relasional seperti yang telah dijelaskan pada bab sebelumnya. Perhatikan contoh di bawah ini!

```
if ((a >= 2) && (b == 3))
      {
      .
      .
      }
```

Jangan Anda tuliskan

```
if (a >= 2) && (b == 3)
      {
      .
      .
      }
```

atau

```
if ((a >= 2) && (b = 3))
      {
      .
      .
      }
```

Perintah b = 3 merupakan assignment bukan relasional.

Catatan penting:
C++ selalu memperlakukan nilai tidak sama dengan nol sebagai TRUE dan nilai nol sama dengan FALSE. Oleh karena itu, dua perintah di bawah ini adalah identik.

```
if (bil % 2 != 0)
      cout << "Bilangan ganjil";

if (bil % 2)
      cout << "Bilangan ganjil"
```

Selain itu, IF juga dapat berbentuk seperti di bawah ini.

```
if (kondisi1)
      statement1;
else if (kondisi2)
      statement2;
else if (kondisi3)
      statement3;
.
.
else statement;
```

Anda tahu kan maksudnya?

Latihan:

Buatlah program dengan C++ untuk menentukan hari menggunakan IF. Desain tampilannya sbb:

```
PROGRAM MENENTUKAN NAMA HARI

Pilihan:
A = SENIN    C = RABU     E = JUM'AT    G = AHAD
B = SELASA   D = KAMIS    F = SABTU

Masukkan Kode Hari (A..G) :
```

Skenario:
Apabila kode hari yang dimasukkan A s/d G, maka selanjutnya akan tampil nama hari sesuai kodenya. Tapi apabila selain 1 s/d 7, maka akan tampil "MAAF KODE HARINYA SALAH". Kode hari harus bisa juga membaca huruf kecil, artinya jika kode harinya 'a', maka akan tampil SENIN, dst.

B. Statement SWITCH

Statement SWITCH juga berfungsi sama dengan IF. Perintah SWITCH sama dengan perintah CASE OF dalam PASCAL.

Sintaks:

```
switch (variabel)
{
case value1 : statement1;
              break;
case value2 : statement2;
              break;
.
.
```

```
default      : statement;    /* optional */
               break;
}
```

contoh penggunaan:

```
#include <iostream.h>
#include <conio.h>
void main()
{
  int bil;
  clrscr();
  cout << "Masukkan bilangan : ";
  cin >> bil
  switch (bil)
  {
  case 1 : cout << "Anda memasukkan bil. satu";
           break;
  case 2 : cout << "Anda memasukkan bil. dua";
           break;
  case 3 : cout << "Anda memasukkan bil. tiga";
           break;
  default: cout << "Anda memasukkan bil selain 1, 2, dan 3";
           break;
  }
}
```

Selanjutnya coba kalian hapus semua break program di atas dan kalian jalankan. Apa yang terjadi?? Keanehan akan muncul. Why ??

Latihan:

Dengan soal yang sama dengan latihan sebelumnya, kalian coba buat program C++ nya menggunakan SWITCH.

C. Statement FOR

Statement FOR digunakan untuk menyatakan perulangan (seperti PASCAL). Sintaksnya:

```
for (ungkapan1; ungkapan2; ungkapan3)
  {
    .
    .
  }
```

- Ungkapan1 merupakan statement awal (inisialisasi)
- Ungkapan2 merupakan kondisi/syarat perulangan dilakukan
- Ungkapan3 merupakan statement control untuk perulangan

Contoh:

```
for (a = 1; a <= 5; a++)
      {
      cout << "Hello world \n";
      }
```

NB: tipe data variabel a adalah integer

Perintah di atas akan menampilkan teks Hello World sebanyak 5 buah. Perhatikan tanda **a++**. Apa maksudnya?

Selain berupa angka, pencacah perulangan juga dapat berupa karakter. Contoh

```
for (huruf = 'Z'; huruf >= 'A'; huruf--)
      {
      cout << "Huruf abjad= " << huruf << "\n";
      }
```

Perintah di atas akan menampilkan teks Huruf abjad = ... mulai dari Z sampai dengan A. Perhatikan perintah **huruf--**

```
for (angka = 1; angka <= 6; angka+=2)
      {
      cout << "Isi dari angka = " << angka << endl;
      }
```

Perintah di atas akan menampilkan angka 1, 3, 5. Kok bisa? Perhatikan perintah **angka+=2**.

Di bawah ini adalah program untuk mencetak bilangan genap yang kurang dari n (n diperoleh dari input).

```
#include <iostream.h>
#include <conio.h>
void main()
{
      int bil, n;
      cout << "Masukkan n = ";
      cin >> n;
      for (bil = 0; bil < n; bil++)
            {
            if (bil % 2 == 0) cout << bil << " ";
            }
}
```

Latihan:

Buatlah program C++ untuk membuat tampilan segitiga seperti di bawah ini menggunakan FOR.

```
*
* *
* * *
* * * *
* * * * *
```

.
.
dst

Inputnya merupakan tinggi segitiga.

D. Statement WHILE

Statement WHILE juga digunakan untuk menyatakan perulangan. Penggunaannya mirip pada PASCAL. Sintaksnya:

```
while (kondisi)
      {
      .
      .
      }
```

contoh:

Dua perintah di bawah ini adalah identik.

```
for (a = 1; a <= 5; a++)
      {
      cout << "Hello world \n";
      }
```

dengan

```
a = 1;
while (a <= 5)
   {
      cout << "Hello world \n";
      a++;
   }
```

> **Penting!!!**
> Jika Anda menggunakan WHILE, pastikan bahwa suatu saat bagian kondisi sampai bernilai FALSE. Apabila tidak, proses perulangan akan terus berjalan selamanya.

Contoh program di bawah ini digunakan untuk menjumlahkan sejumlah data angka. Angka yang akan dijumlahkan diinputkan satu-persatu. Proses pemasukan data angka akan berhenti ketika dimasukkan angka –1. Setelah itu tampil hasil penjumlahannya.

```
#include <iostream.h>
#include <conio.h>
void main()
{
      int data, jumlah,cacah;
      jumlah = 0;
      data = 0;
      cacah = 0;
      while (data != -1)
```

```
    {
        cout << "Masukkan data angka : ";
        cin >> data;
        jumlah += data;
        cacah++;
    }
    cout << "Jumlah data adalah : " << jumlah" << endl;
    cout << "Rata-rata : " << jumlah/cacah;
}
```

Kondisi dalam WHILE juga dapat menggunakan operator logika. Misalnya

```
while ((kondisi1) && (kondisi2))
    {
    .
    .
    }
```

Latihan:

Buatlah simulasi menu program dengan tampilan di bawah ini menggunakan WHILE.

MENU PILIHAN
1. Baca Data
2. Ubah Data
3. Hapus Data
4. Exit

Pilihan Anda (1/2/3/4) ? ...

Apabila dipilih menu no 1, maka akan tampil teks "Anda memilih menu 1". Demikian pula untuk menu 2 dan 3. Kemudian setelah itu muncul teks "Tekan ENTER untuk kembali ke menu utama". Artinya begitu kita tekan ENTER menu pilihan akan muncul kembali, dst. Akan tetapi bila yang dipilih menu 4 (EXIT), program langsung berhenti.

E. Statement DO ... WHILE

Perintah DO ... WHILE hampir sama dengan WHILE sebelumnya. Sintaknya:

```
do
  {
  .
  .
  }
while (kondisi);
```

Perbedaan dengan WHILE sebelumnya yaitu bahwa pada DO WHILE statement perulangannya dilakukan terlebih dahulu baru kemudian di cek kondisinya. Sedangkan WHILE kondisi dicek dulu baru kemudia statement perulangannya dijalankan. Akibat dari hal ini adalah dalam DO WHILE minimal terdapat 1x

perulangan. Sedangkan WHILE dimungkinkan perulangan tidak pernah terjadi yaitu ketika kondisinya langsung bernilai FALSE.

Contoh:

```
a = 1;
do
   {
     cout << "Hello world \n";
     a++;
   }
while(a==0)
```

Perintah di atas akan muncul satu buah Hello World. Bandingkan dengan yang berikut ini:

```
a = 1;
while(a==0)
  {
     cout << "Hello world \n";
     a++;
  }
```

Perintah di atas sama sekali tidak menampilkan Hello World, karena kondisinya langsung FALSE.

TUGAS:

1. Buatlah program dengan C++ untuk menampilkan semua penyelesaian dari persamaan x + y + z = 20.
   Dengan x, y, z bilangan bulat >= 0.

   Contoh tampilan outputnya:

   ```
   -----------------------------
   x            y         z
   -----------------------------
   0            0         20
   0            1         19
   0            2         18
   ```

   dst

2. Seseorang punya rekening tabungan di bank sebesar Rp. 10.000,- (saldo awal). Selanjutnya ia dapat menyetor atau mengambil tabungannya. Buatlah program dengan C++ untuk keperluan transaksi di bank tsb. Tampilan menu utamanya sbb:

   PT. BANK SYARIAH BENERAN

   Saldo : ....

   Menu Transaksi

   1. Setor Tabungan
   2. Ambil Tabungan
   3. Exit

   Pilihan menu (1/2/3) ? ...

   Ketentuan:
   Bank membuat kebijakan bahwa saldo minimum yang harus disisakan di rekening adalah Rp. 10.000,-